REMARKS

Applicant appreciates the time taken by the Examiner to review Applicant's present application. This application has been carefully reviewed in light of the Official Action mailed October 7, 2008.

Claims Status

Claims 1-8, 12-18 and 23-26 were pending and rejected. Claims 1, 12 and 23 are amended herein. Support for the amendments presented herein can be found in the specification as originally filed. *See e.g.,* Specification, paras. 50-51, 53-60, 65, 70-73, 81, 90, 92 and 96-105. No new matter is introduced. Thus, claims 1-8, 12-18 and 23-26 remain pending. Applicant respectfully requests reconsideration and favorable action in this case.

Examiner Interview Summary

A telephonic interview was conducted on January 13, 2009, between Examiner Cheryl Lewis, inventors Clint Miller, Mark Castoe and Ray Renteria, Attorney Ari Akmal, and Agent Kevin Gust. During the interview, embodiments of the claims, a proposed amendment and the cited prior art were discussed. Examiner Lewis agreed to allow the claims if independent claims 1, 12 and 23 are amended to contain similar language as the proposed amended claim. Applicant appreciates the time and effort taken by Examiner Lewis to review Applicant's present application and to discuss the pending claims and the cited prior art.

Rejections Under 35 U.S.C. § 102

Claims 1-5, 8, 12-16 and 23-26 were rejected under 35 U.S.C. §102(a) as being anticipated by U.S. Patent Application Publication No. 2006/0167927 ("Edelstein"). Applicant traverses the rejection. Claims 12 and 23 contain similar language as claim 1. Accordingly, traversal of the rejection will be collectively addressed as it pertains to claim 1.

Claim 1, as amended, recites:

> A method for searching an applied data model, comprising:
> on a computer having a computer memory and a processor, translating a query to a set of statements operable to search the applied data model to an arbitrary level,

wherein the applied data model comprises a plurality of defined data structures, wherein each of the data structures comprises one or more fields or properties associated with the data structure, wherein all data structures of the same type contain the same properties, wherein the applied data model comprises at least one component and a relationship corresponding to the at least one component,

wherein the at least one component represents a physical or logical entity in the arbitrarily complex environment, wherein each component has a set of fields which contains information relating to an atomic entity associated with the at least one component, wherein the set of fields comprises:

a set of property fields containing information about the attributes or characteristics of the component; and

a field that contains a link to its component type,

wherein values assigned to the properties in the component are based on the attributes of the entity which the component was instantiated to represent,

wherein the relationship represents an association between the physical or logical entity and other physical or logical entities in the arbitrarily complex environment, wherein the relationship comprises:

a field that is a foreign key to its relationship type; and

a set of property fields containing information about one or more of the attributes of the relationship,

wherein the components are stored in a schema, wherein property definitions of each component are linked to a type of component, wherein changes made to the type of component are automatically associated with all components of that type of component without changing the schema to reflect a corresponding change in the arbitrarily complex environment,

wherein the schema is implemented in a database,

wherein the query is a component query or a relationship query;

searching the applied data model to the arbitrary level based on the set of statements translated from the query,

wherein the query is in a first query language, and

wherein the set of statements is capable of execution by a database management system supporting a second query language, wherein the database management system is associated with the database;

producing a set of replies to the set of statements, wherein the set of replies includes at least one component or one relationship at the arbitrary level; and

processing the set of replies according to the query.

Thus, embodiments of a method for searching an applied data model may include translating a query to a set of statements operable to search the applied data model to an arbitrary level, searching the applied data model to the arbitrary level based on the set of statements translated from the query, producing a set of replies to the set of statements, wherein the set of replies includes at least one component or one relationship at the arbitrary level, and processing the set of replies according to the query. The data model may include a plurality of defined data structures, wherein each of the data structures comprises one or more

fields or properties associated with the data structure, wherein all data structures of the same type contain the same properties, and wherein the applied data model comprises at least one component and a relationship corresponding to the at least one component. Further, in some embodiments, at least one component represents a physical or logical entity in the arbitrarily complex environment, wherein each component has a set of fields which contains information relating to an atomic entity associated with the at least one component, wherein the set of fields comprises a set of property fields containing information about the attributes or characteristics of the component, and a field that contains a link to its component type, wherein values assigned to the properties in the component are based on the attributes of the entity which the component was instantiated to represent. Further, in some embodiments, the relationship represents an association between the physical or logical entity and other physical or logical entities in the arbitrarily complex environment, wherein the relationship comprises a field that is a foreign key to its relationship type and a set of property fields containing information about one or more of the attributes of the relationship. In some embodiments, the components are stored in a schema, wherein property definitions of each component are linked to a type of component, wherein changes made to the type of component are automatically associated with all components of that type of component without changing the schema to reflect a corresponding change in the arbitrarily complex environment, and wherein the schema is implemented in a database. In some embodiments, the query is a component query or a relationship query, the query is in a first query language, and wherein the set of statements is capable of execution by a database management system supporting a second query language, wherein the database management system is associated with the database.

As mentioned in the Examiner interview conducted January 13, 2009, embodiments may be useful for querying a schema corresponding to a data model modeling an arbitrarily complex environment, in which the schema may reflect changes to the environment without rewriting significant amounts of code. For example, a server computer may have a set of fields which are generic to all server computers. The fields in the component representing the server component may have selected values to associate the component with a particular server computer, associate the component with a particular type of component, contain properties based on the type of component, and may be assigned values corresponding with the particular server. If any of the information must be changed to reflect a change in the arbitrarily complex environment, only changes to the information contained in the fields are changed – the entire component does not need to be re-coded. Similarly, relationships may be built programmatically to reflect changes in the arbitrarily complex environment. The property fields may allow attributes of the

relationships to be represented by a name and value pair. In one particular example, a first field may be set to a name corresponding to a first component associated by the relationship and a second field may be set to a name corresponding to a second component. The relationship may have a field that links the relationship to a relationship type. The data model may be stored in a schema, in for example, a table. If either component is renamed, the relationship does not need to be altered (i.e, re-coded), and the arbitrarily complex environment is still modeled, by, for example, changing the name in the table. An advantage of this may be the ability to efficiently perform queries against the schema. A set of SQL statements that was performed on an existing schema may still be performed for the schema when changes are made to the data model.

In contrast, prior art approaches to modeling an entity-relationship of an environment are generally hard-coded and any changes to the arbitrarily complex environment would be time-consuming to change in the schema. As such, prior art attempts to store information about components generally requires some foreknowledge of the kinds of objects that the schema is intended to model. An extension of this concept is that queries performed by prior art methods generally required some foreknowledge of the kinds of objects to receive, and any changes to the arbitrarily complex environment may require modifications to the SQL statements to effect explicit changes to the query.

Regarding Edelstein, Edelstein teaches a method for data query and location through a central ontology model. Edelstein describes enterprise data systems as typically having multiple data sources with different data schemas, with various databases conforming to various data schemas. (See, Edelstein, para. 11.) A data locator may be used to receive a list of constructs corresponding to the data of interest and the various data sources containing data for such constructs. (See, Edelstein, paras. 15-17.) An ontology query may include a first clause indicating at least one subject class to which the query applies, and a second clause indicating at least one property or composition of properties defined on the at least one subject class.

In the rejection, the Examiner states that Edelstein anticipates translating a query into a set of statements operable to search the applied data model to an arbitrary level wherein the applied data model is a representation of an arbitrarily complex environment. Applicant respectfully submits that Edelstein does not teach or describe a method that can accommodate changes to the arbitrarily complex environment. In general, Edelstein describes methods and systems that follow the prior art approach as described above. Thus, Edelstein appears to require some foreknowledge of objects or the kinds of objects to be retrieved in order to retrieve the objects, and explicit changes to the queries are necessary when changes are made to the schema.

For at least the foregoing reasons, Applicant respectfully submits that Edelstein fails to anticipate one or more of translating a query to a set of statements operable to search the applied data model to an arbitrary level, wherein the applied data model comprises a plurality of defined data structures, wherein each of the data structures comprises one or more fields or properties associated with the data structure, wherein all data structures of the same type contain the same properties, wherein the applied data model comprises at least one component and a relationship corresponding to the at least one component, wherein the at least one component represents a physical or logical entity in the arbitrarily complex environment, wherein each component has a set of fields which contains information relating to an atomic entity associated with the at least one component, wherein the set of fields comprises a set of property fields containing information about the attributes or characteristics of the component; and a field that contains a link to its component type, wherein values assigned to the properties in the component are based on the attributes of the entity which the component was instantiated to represent, wherein the relationship represents an association between the physical or logical entity and other physical or logical entities in the arbitrarily complex environment, wherein the relationship comprises a field that is a foreign key to its relationship type; and a set of property fields containing information about one or more of the attributes of the relationship, wherein the components are stored in a schema, wherein property definitions of each component are linked to a type of component, wherein changes made to the type of component are automatically associated with all components of that type of component without changing the schema to reflect a corresponding change in the arbitrarily complex environment, wherein the schema is implemented in a database, wherein the query is a component query or a relationship query, searching the applied data model to the arbitrary level based on the set of statements translated from the query, wherein the query is in a first query language, and wherein the set of statements is capable of execution by a database management system supporting a second query language, wherein the database management system is associated with the database, producing a set of replies to the set of statements, wherein the set of replies includes at least one component or one relationship at the arbitrary level, and processing the set of replies according to the query, as recited in Claim 1. Accordingly, withdrawal of this rejection is requested.

Rejections under 35 U.S.C. § 103

Claims 6-7 and 17-18 were rejected under 35 U.S.C. 103(a) as being unpatentable over Edelstein as applied to claims 5 and 16 above, and further in view of U.S. Patent No. 6,509,898 ("Chi"). The rejection is traversed. Applicant respectfully submits that claims 6-7 and 17-18

depend from independent claims 1 and 12, respectively, which the Examiner agreed to allow in the Examiner interview conducted January 13, 2009. Applicant respectfully submits that claims 6-7 and 17-18 are thus non-obvious in view of Edelstein or Chi. Accordingly, withdrawal of this rejection is respectfully requested.

CONCLUSION

Applicant has now made an earnest attempt to place this case in condition for allowance. Other than as explicitly set forth above, this reply does not include any acquiescence to statements, assertions, assumptions, conclusions, or any combination thereof in the Office Action. For the foregoing reasons and for other reasons clearly apparent, Applicant respectfully requests full allowance of Claims 1-8, 12-18 and 23-26. The Examiner is invited to telephone the undersigned at the number listed below for prompt action in the event any issues remain.

An extension of 1 (one) month is requested and a Notification of Extension of Time Under 37 C.F.R. § 1.136 with the appropriate fee is enclosed herewith.

The Director of the U.S. Patent and Trademark Office is hereby authorized to charge any fees or credit any overpayments to Deposit Account No. 50-3183 of Sprinkle IP Law Group.

Respectfully submitted,

Sprinkle IP Law Group
Attorneys for Applicant

Ari G. Akmal
Reg. No. 51,388

Date: January 16, 2009
1301 W. 25th Street, Suite 408
Austin, TX 78705
Tel. (512) 637-9220
Fax. (512) 371-9088